Data Mining for Features Using Scale-Sensitive Gated Experts

Ashok N. Srivastava, Renjeng Su, *Fellow*, *IEEE*, and Andreas S. Weigend

Abstract—This article introduces a new tool for exploratory data analysis and data mining called *Scale-Sensitive Gated Experts* (SSGE) which can partition a complex nonlinear regression surface into a set of simpler surfaces (which we call features). The set of simpler surfaces has the property that each element of the set can be efficiently modeled by a single feedforward neural network. The degree to which the regression surface is partitioned is controlled by an external scale parameter. The SSGE consists of a nonlinear gating network and several competing nonlinear experts. Although SSGE is similar to the mixture of experts model of Jacobs et al. [10] the mixture of experts model gives only one partitioning of the input-output space, and thus a single set of features, whereas the SSGE gives the user the capability to discover families of features. One obtains a new member of the family of features for each setting of the scale parameter. In this paper, we derive the Scale-Sensitive Gated Experts and demonstrate its performance on a time series segmentation problem. The main results are: 1) the scale parameter controls the granularity of the features of the regression surface, 2) similar features are modeled by the same expert and different kinds of features are modeled by different experts, and 3) for the time series problem, the SSGE finds different regimes of behavior, each with a specific and interesting interpretation.

Index Terms—Mixture of experts, mixture model, classification and regression, time series segmentation, neural networks.

1 Introduction

We give an algorithm which learns to carve the joint input-output space into partially overlapping regions depending on the magnitude of a scale parameter and then builds a local model for each feature. Other models such as the mixture of experts model suggested by Jacobs et al. [10] do not allow for an external adjustment of the strength of associating an input-output pair to a local model. The scale-sensitive gated experts (SSGE) implicitly allows for a hierarchy of features to develop: global features (which correspond to small values of the scale parameter) subsume local features (which correspond to large values of the scale parameter). The features may be complex nonlinear surfaces from disjoint regions in the input-output space and are modeled by a set of expert networks, whose task is to predict the value at the regression surface given the input, and a gate network, whose task is to learn to associate inputs with particular experts.

The sensitivity of the algorithm to the scale of features in the input-output space is governed by the scale parameter. For small values of the scale parameter, global features are extracted, whereas for large values of the scale parameter, local features are extracted. Thus, the scale parameter defines the level of coarseness, or *granularity* of the features that the algorithm extracts. We call the process of sweeping from global to local features *feature refinement*. The scale parameter arises naturally in the derivation of the model: we do not arbitrarily add a parameter to the model. The algorithm is governed by an important quantity called the *association probability* which governs the probability of associating an input-output pair with a local model or expert. The association probability is parametrized by the scale parameter and is derived by making very general assumptions about the data.

The intended application area for this algorithm is in exploratory data analysis and data mining. In these fields, the characteristics of a correct or optimal solution is often not known, and the analyst must systematically search through a series of solutions to understand the nature of the data space. As one views the results for different values of the scale parameter, a better understanding of the complexity of the data space often results.

1.1 Structure of Article

Section 2 discusses the application of SSGE to time series segmentation problems. These problems motivate the development of this algorithm. Section 3 derives the SSGE association probabilities using the principle of maximum entropy and interprets them as a function of the scale parameter. We compare the association probabilities derived here with those obtained in the standard gated experts architecture.

Section 4 derives the corresponding cost function. We prove that the minimum of this cost function corresponds to the most probable set of associations. Next, parameter update rules are given for the nonlinear and linear case. The following section demonstrates the SSGE on a time series segmentation problem: a computer generated time series which undergoes regime switches. Section 7 summarizes the Scale-Sensitive Gated Experts and suggests future areas of research.

- *A.N. Srivastava is with the Deep Computing Consulting Group, IBM Almaden Research Center, San Jose, CA 95120. E-mail: ashoks@almaden.ibm.com*
- *R. Su is with the Department of Electrical and Computer Engineering, University of Colorado, Boulder, CO 80309-0529. E-mail: sur@colorado.edu.*
- *A.S. Weigend is with Emotioneering, Inc., 2260 Forestview Ave., Hillsborough, CA 94010. E-mail andreas@weigend.com.*

Manuscript received 6 May, 1998; revised 8 Sept. 1999. Recommended for acceptance by I. Sethi. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107836.

1.2 Related Work

The SSGE has intimate ties with two other model classes: mixture models [10] and clustering models [16], [6], [13]. In this section we briefly discuss these two model classes and compare with the SSGE.

Jacobs et al. [10] introduced mixture models to the connectionist community, where the output of the system is a weighted sum of expert outputs and each expert is a regression model. The weights sum to one and indicate the probability that a particular expert is appropriate given the input. These models were subsequently developed into *hierarchical mixtures of linear experts* by Jordan and Jacobs [11]. Weigend et al. [22] applied the mixture of experts model to time series problems. These mixture models do not give the user any control over the degree that the input-output space is partitioned.

Rose et al. [16] give a method to perform clustering while giving the user control over the granularity of the clusters. They introduce a method called *Thermodynamic Clustering*, where the probability of assigning a particular data point to a cluster is a function of a scale parameter. For small values of the parameter, this model allows a data point to be captured by more than one cluster, thus allowing for soft-clustering. On the other hand, for large values of the parameter, the model forces a hard-clustering: a data point can only be assigned to a single cluster [23].

Related work from the connectionist community include Durbin and Willshaw [4] which is a special case of gated experts with the variance annealed. They applied their architecture to the Traveling Salesman Problem. Pawelzik et al. [13] and Fancourt and Principe [5] applied the annealed variance gated experts to time series problems. An important distinction in the former case from the present is that the gate network in Pawelzik et al. [13] is not a function of the input. Also see Jacobs and Jordan [9] for applications to control.

The SSGE represent a marriage between the idea of mixture models and the idea of thermodynamic clustering. Interesting comparisons between the gated experts models discussed here and the hidden markov model can be found in Shi [18].

2 Applications to Time Series Segmentation Problems

The SSGE can be applied to any data analysis problem where input-output data are available. Consider, for example, the problem of predicting a univariate time series $\{d^t\}_{t=1}^{T}$. A standard prediction method relies on the concept of embedding [20], where the next value in the series, d^t is expressed as a function of the last p values, $[d^{t-1}, \ldots, d^{t-p}]$. The lagged values form a set of inputs and the values to be predicted form a set of outputs. The prediction problem is defined as learning a regression surface which maps the inputs to the outputs. The SSGE learns this regression surface and also partitions the surface into different regions depending on the setting of the scale parameter.

In this paper, we use the SSGE to analyze multistationary time series, i.e., time series which arise from a data generating process that switches its mode of behavior. This switch could manifest as a shift in the mean, variance, or some other statistic and it indicates that the underlying dynamics of the data generating process has changed. We assume that the change in regime is observable in the time series, and thus will appear in the embedding space as a variation in the regression surface.

A key problem in the analysis and prediction of such systems is to identify these so-called *regime shifts*: when a shift occurs and what quantity changed. The process of identifying the times at which a shifts occur is known as time series segmentation. These segments could be short time intervals compared to the time scale of the series or relatively long time intervals. In either case, the segments that we consider are intervals of arbitrary duration in time. Basseville and Nikiforov [1] give an excellent review of methods for predicting and detecting regime shifts. Weigend et al. [22] applied the mixture of experts model to the double nonlinear case, where the both the gate and expert networks are nonlinear feed-forward neural networks to time series problems. The statistics community first introduced the idea of modeling a regime shift [14] by assuming a mixture model where the output is a weighted sum of expert outputs and the weights sum to unity and indicate the probability that the system is dwelling in a particular regime.

To apply the SSGE to time series segmentation, we make the following assumptions:

- We assume that the next value of the time series can be expressed as a nonlinear combination of past values and other relevant quantities. This assumption allows us to model the time series as a regression problem where the task is to learn a potentially nonlinear mapping from inputs to outputs.
- We assume that the dynamics of the time series is unknown and that it must be inferred from the input-output mapping.
- We assume that the segmentation is unknown apriori and that it can be inferred from the input-output data.

For many real-world time series segmentation problems, the analyst often does not know whether or not there is a correct segmentation, and how many different segments (or regimes) exist. To our knowledge, for a given model configuration, current time series segmentation procedures give a single segmentation without giving the analyst any other possible segmentations. The SSGE is a tool especially designed to give the user the ability to sweep through a wide range of possible segmentations, after which the user can choose a segmentation that matches the task at hand.

The SSGE attacks the time series segmentation problem by:

- Computing the probability that an input-output pair arose from a particular regime. Since each regime is modeled by a single expert, this probability is equivalent to the probability of associating an input-output pair to a particular expert.
- This so-called association probability is a function of an external scale-parameter which governs the strength of the association. These associations, in

turn, produce the segmentation of the time series in the sense that a plot of the association probabilities as a function of time indicates the segmentation.
- Learning the underlying dynamics of each segment via local nonlinear regression (expert networks).
- Learning to predict the association probabilities from the input alone (gate network), and not relying on the output. This is necessary because the output is unavailable during model testing and verification.

3 Development of the Scale-Sensitive Gated Experts

The SSGEs operation is governed by a quantity called the association probability, which is the probability of associating an input-output pair to a particular expert (or local model). Several local models can share a given input-output pair, thus yielding a soft classification. The association probability is a function of the error between an expert's prediction of the output (given the input) and the actual output value. This probability is also a function of an external parameter β that adjusts the strength of the associations and thus the coarseness of the features. In this section, we derive the association probabilities.

We begin by defining the variables we use:

- $\mathbf{x}$ is the input vector
- d is the target (or "desired output value")
- $y_j(\mathbf{x})$ is the output of expert j (corresponds to the mean of the Gaussian). We assume a univariate model although the theory readily generalizes to multivariate outputs.
- σ_j is the standard deviation of the Gaussian represented by expert j
- $P(Y = y \mid \mathbf{x}, j)$ is the probability density associated with the jth expert for the stochastic variable Y to take the value y
- $g_j(\mathbf{x})$ is the output of the gating network, denoting the probability that a given pattern is generated by the jth expert, given the input $\mathbf{x}$; i.e., $g_j^t = P(s^t = j\mathbf{x}^t)$
- $H_j(\beta, \mathbf{x}, d, y_j)$ is the posterior probability of the jth expert, given the output y_j and the pattern, i.e., input $\mathbf{x}$, target d. This is also called the *association probability*, or the probability of associating an input-output pair to a particular expert,
- β denotes the scale parameter,
- $s^t = j$ denotes the event that the tth pattern is generated by the jth expert ($1 \leq j \leq K$)
- t is the pattern index
- T is the total number of patterns
- Θ_j and Θ_g denote the set of parameters of expert j and the gate, respectively.

For notational simplicity, in many of the equations to follow, we supress the explicit dependence of the variables on the parameters and inputs. Thus, instead of writing $g_j(\mathbf{x}^t, \Theta_g)$, we may write g_j^t.

3.1 Derivation of the Association Probability

Suppose we define a per-pattern error function

$$E_j^t = -\log(P(s^t = j, d^t \mid \mathbf{x}^t),$$

which denotes the complete negative log-likelihood of the data given the input. The errorfunction E_j^t denotes the cost of associating the tth input-output pair to the jth expert. We expand the error function to obtain

$$E_j^t = -\log(P(s^t = j, d^t \mid x^t) \tag{1}$$

$$= -\log(P(s^t = j \mid \mathbf{x}^t)P(d^t \mid s^t = j, \mathbf{x}^t)) \tag{2}$$

$$= -\log(g_j^t P(d^t \mid s^t = j, \mathbf{x}^t)). \tag{3}$$

Thus, for a given j, the error function factors into two additive terms: a classification error and a regression error:

$$E^t = E_C^t + E_R^t \tag{4}$$

The classification error $E_C^t = -\log P(s^t = j \mid \mathbf{x}^t)$ is the negative log likelihood of choosing a particular expert given the input. The regression error $E_R^t = -\log P(d^t \mid s^t = j, \mathbf{x}^t)$ is related to the probability of observing the desired value d^t given the input and the choice of the jth expert.

To obtain the association probabilities, we use the principal of maximum entropy for the following reasons:

- We have no model for the distribution of the *correct* association probabilities.
- We have two constraints, namely that the average error of the model is finite, and that the sum of the probabilities is equal to unity.

Given these constraints, the most likely model for the association probabilities is the one whose distribution is closest to a uniform distribution, which is the solution to the maximum entropy problem.

To obtain the association probabilities using a maximum entropy framework, we have the following optimization problem to solve. We desire to maximize the entropy

$$\mathcal{S} = -\sum_{t=1}^{T}\sum_{j=1}^{K} H_j^t \log H_j^t \tag{5}$$

subject to the following two constraints:

$$\sum_{j=1}^{K} H_j^t = 1 \ \forall\ t \tag{6}$$

$$< E >= \sum_{t=1}^{T}\sum_{j=1}^{K} E_j^t H_j^t. \tag{7}$$

The association probability is computed according to the principle of maximum entropy to avoid making further assumptions about the nature of the distribution. We solve this problem using the standard theory of Lagrange multipliers and obtain the "canonical" or Gibbs distribution which is parametrized by a scale parameter β [7], [19], [2]:

$$H_j^t(\beta) = \frac{\exp(-\beta E_j^t)}{\sum_{k=1}^{K}\exp(-\beta E_k^t)}. \tag{8}$$

Assuming a Gaussian noise model, the final result is:

$$H_j^t(\beta) = \frac{\exp(-\beta E_j^t)}{\sum_{k=1}^{K} \exp(-\beta E_k^t)} \tag{9}$$

$$= \frac{\left(g_j^t P(d^t \mid s^t = j, x^t)\right)^\beta}{\sum_{k=1}^{K} \left(g_k^t P(d^t \mid s^t = k, x^t)\right)^\beta} \tag{10}$$

$$= \frac{(g_j(\mathbf{x}^t, \theta_g))^\beta \left[\frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{\left(d^t - y_j(\mathbf{x}^t, \theta_j)\right)^2}{2\sigma_j^2}\right)\right]^\beta}{\sum_{k=1}^{K} (g_k(\mathbf{x}^t, \theta_g))^\beta \left[\frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(d^t - y_k(\mathbf{x}^t, \theta_k))^2}{2\sigma_k^2}\right)\right]^\beta}. \tag{11}$$

3.2 Interpretation of the Association Probability and Scale Parameter

We now discuss the effect β has on the association probabilities. Consider the ratio of the probabilities of associating the tth pattern with two different experts, j, and k:

$$Q(\Delta E) = \frac{P(s^t = j \mid x^t, d^t)}{P(s^t = k \mid x^t, d^t)} = \exp(\beta(E_j^t - E_k^t)). \tag{12}$$

For small β, the difference in error between two states is reduced. Thus, a pair (x^t, d^t) is easily associated with more than one expert. On the other hand, for large β, the difference in error between two states gets magnified and so a pair (x^t, d^t) is associated with that expert which minimizes the error. Thus, the scale parameter β:

- adjusts the probability of associating an input-output pair to a particular local model;
- sets the strength of association. Larger values bias the model towards a "binary" configuration, where only one expert is used to model the data; and
- naturally arises from the maximum entropy formulation as a Lagrange multiplier.

3.3 Comparison with Gated Experts

Now that we have an equation for the association probabilities, we can analyze it and compare it to association probabilities derived for the Gated Experts [22],

$$h_j^t = h_j\left(\mathbf{x}^t, d^t, y_j(\mathbf{x}^t, \theta_j), g_j(\mathbf{x}^t, \theta_g)\right) \tag{13}$$

$$= \frac{g_j(\mathbf{x}^t, \theta_g) \ \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{\left(d^t - y_j(\mathbf{x}^t, \theta_j)\right)^2}{2\sigma_j^2}\right)}{\sum_{k=1}^{K} g_k(\mathbf{x}^t, \theta_g) \ \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(d^t - y_k(\mathbf{x}^t, \theta_k))^2}{2\sigma_k^2}\right)}. \tag{14}$$

We find that (14) and (11) are identical, except for the parameter β, and that if we take $\beta = 1$ in (11) we obtain the same association probabilities. Thus, the maximum entropy case reduces to the maximum likelihood case for $\beta = 1$. The parameter β indicates our prior assumption on the probabilities. Fig. 1 indicates the effect of β on the association probability given in (14).

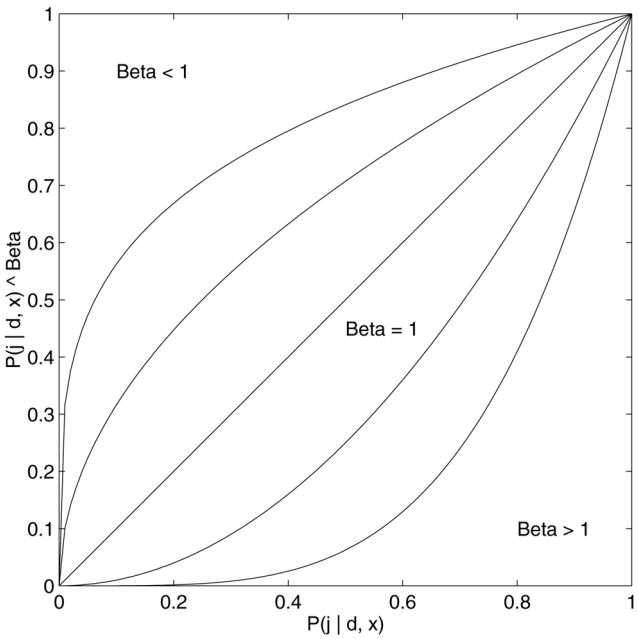

Fig. 1. This figure indicates the effect of β on the association probability. For $\beta < 1$, the association probability is emphasized, thus increasing the chance that two experts share a pattern. For $\beta = 1$ the association probability is unchanged, and for $\beta > 1$, the probability is de-emphasized, and therefore reduces the chance that two experts share a pattern.

We note that the equation given in (14) is a standard mixture equation that arises in a number of sciences, including fuzzy logic, statistical physics, and neural networks. See [8] for a good discussion on this equation in other domains.

4 Obtaining the Most Probable Associations: Derivation of the Cost Function

4.1 Maximizing the Association Probabilities

Given the method to compute the association probabilities $H_j^t(\beta)$, we turn our attention to the problem of computing the parameters which maximize the association probability. This cost function turns out to be nothing other than the thermodynamic free energy.

Suppose we have a set of parameters

$$\Theta = \{\theta_j\} \ \forall \ j = 1 \ldots K$$

and we wish to maximize the probability of these parameters given the data. We can follow a maximum likelihood framework [15], [3], [12], [17] in order to compute the cost function. Instead, we follow a maximum entropy derivation which closely follows those given in [16]. We introduce a set of indicator variables:

$$I_j^t = \begin{cases} 1 & \text{if pattern } t \text{ is generated by the } j\text{th expert} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

that identify which regime the tth pattern belongs to. Taking the set of indicator variables for all patterns and regimes $I = \{I_j^t\}$ and assuming an error function E_j^t, the

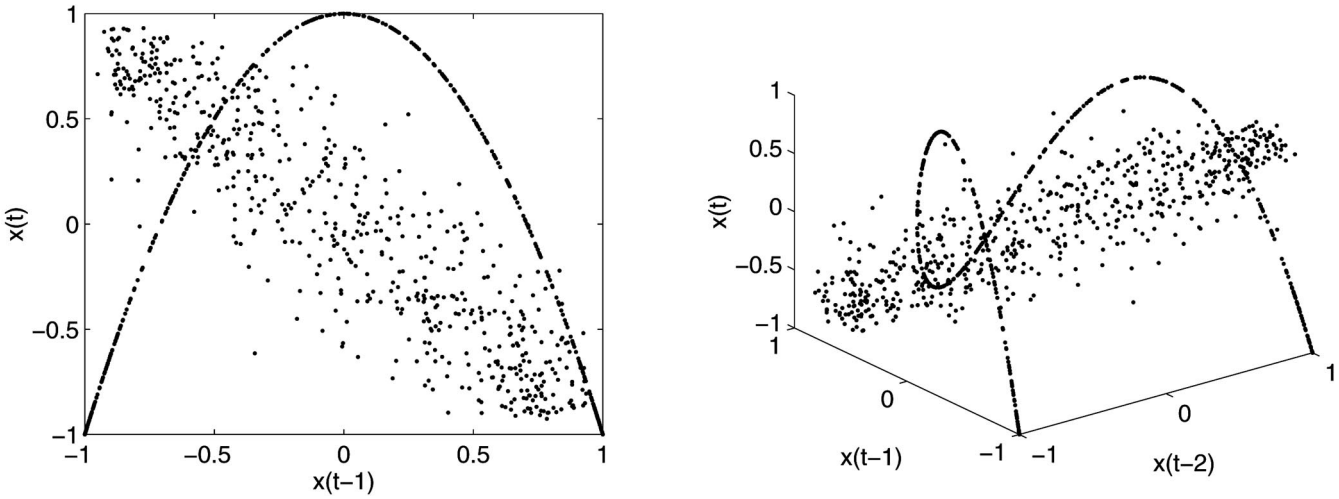

Fig. 2. This figure shows a return plot for the synthetic switching process. Although the two regimes seem overlapping, the three dimensional return plot in the second panel indicates that they are separable. (Reprinted with permission, Weigend et al. [22]).

total error for all patterns and associations is defined as $D(\Theta, I)$, and is given by:

$$D(\Theta, I) = \sum_t \sum_j I_j^t E_j^t. \tag{16}$$

We compute the joint probability $P(\Theta, I)$, which is the probability of observing a model Θ and the regimes I together, using the maximum entropy framework discussed earlier and obtain the following Gibbs distribution:

$$P(\Theta, I) = \frac{\exp(-\beta D(\Theta, I))}{\sum_\theta \sum_I \exp(-\beta D(\theta, I))} \tag{17}$$

$$= \frac{\exp(-\beta D(\Theta, I))}{\Xi}. \tag{18}$$

Where Ξ is the denominator of (17).

Our goal is to compute the most likely set of parameters; the parameters which maximize these probabilities yields the most likely set of associations. We, therefore, need to maximize the probability $P(\Theta)$.

$$P(\Theta) = \sum_I P(\Theta, I). \tag{19}$$

To obtain this distribution, we marginalize the distribution given in (17) with respect to the indicator variable.

This sum is taken over *all possible* associations, where a "possible" association is defined as one in which a only single expert generates an output d^t. Thus, this assumption voids the possibility of more than one expert predicting an output. For example, for the tth pattern, the sum is taken over each of the K possible associations:

$$I_1^t = 1, I_j^t = 0 \ \forall \ j \neq 1 \tag{20}$$

$$I_2^t = 1, I_j^t = 0 \ \forall \ j \neq 2 \tag{21}$$

$$\vdots \tag{22}$$

$$I_K^t = 1, I_j^t = 0 \ \forall \ j \neq K \tag{23}$$

The distribution $P(\Theta)$ is computed by the following straightforward computations:

$$P(\Theta) = \sum_i P(\Theta, I) \tag{24}$$

$$= \frac{1}{\Xi} \sum_I \exp(-\beta D(\Theta, I)) \tag{25}$$

$$= \frac{1}{\Xi} \sum_I \exp(-\beta \sum_t \sum_k I_k^t E_k^t) \tag{26}$$

$$= \frac{1}{\Xi} \sum_I \prod_t \exp(-\beta \sum_k I_k^t E_k^t) \tag{27}$$

$$= \frac{1}{\Xi} \prod_t \sum_k \exp(-\beta E_k^t) \tag{28}$$

$$= \frac{1}{\Xi} Z(\Theta) \tag{29}$$

$$= \frac{Z(\Theta)}{\sum_\theta Z(\Theta)}. \tag{30}$$

The last equality arises from inspection of (17). Choosing a function F as:

$$F = -\frac{1}{\beta} \log Z(\Theta). \tag{31}$$

allows and substituting this expression into (30), we obtain the important relation:

$$P(\Theta) = \frac{Z(\Theta)}{\sum_\theta Z(\Theta)} \tag{32}$$

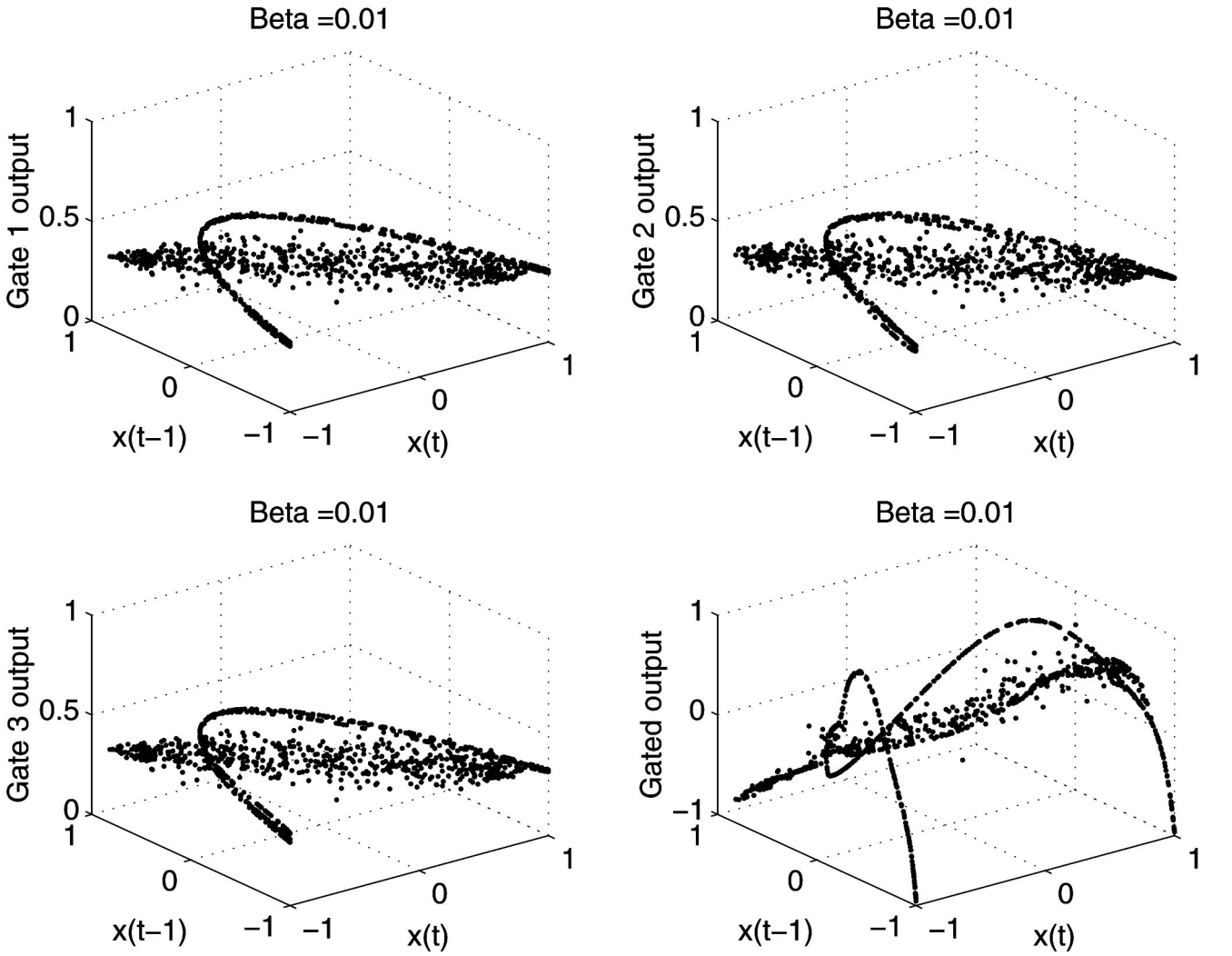

Fig. 3. Gate output for $\beta = 0.01$. The top panel and the plot on the lower left each correspond to an output of the gate network. The plot on the lower right shows the output of the entire SSGE model. Notice that each expert has an activation of approximately $\frac{1}{3}$, which shows that there is no specialization.

$$= \frac{\exp(-\beta F)}{\sum_i \exp(-\beta F)}. \tag{33}$$

This equation shows that to obtain the most likely set of parameters, we need to minimize the function F, which is known as the free energy in statistical mechanics. The expression for the free energy for the Scale-Sensitive Gated Experts is:

$$F = -\frac{1}{\beta}\sum_{t=1}^{T} - \ln \left[\sum_{j=1}^{K} g_j(\mathbf{x}^t, \theta_g)^{\beta} \left(\frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{\left(d^t - y_j(\mathbf{x}^t, \theta_j)\right)^2}{2\ \sigma_j^2}\right)\right)^{\beta}\right]. \tag{34}$$

We estimate the parameters $\theta_g, \theta_1, \theta_2, \cdots, \theta_K, \sigma_1, \sigma_2, \cdots, \sigma_K$ by minimizing the free energy function F with respect to the parameters.

The derivation of the free energy above assumes that a single local model is responsible for a particular input-output pair. We know that for large β, the association probabilities behave in a winner-take-all manner and, thus, the assumption is satisfied. For moderate or small values of the scale parameter, however, (12) indicates that more than one expert can share an input-output pattern.

4.2 Derivation of the Parameters Updates for the Nonlinear Case

The SSGE model is a nonlinear model: Thus, we cannot obtain analytical solutions for the optimal values of the parameters of the gate and expert networks. Instead, we obtain a weight update rule and use a method such as gradient descent or BFGS to optimize the cost function.

We give the weight update rules and discuss some implications. The weight update rules are computed by taking the gradient of the cost function F with respect to the parameters. For the expert network, we have:

$$\frac{\partial F}{\partial \theta_j} = \sum_{t=1}^{T} H_j^t(\beta) \frac{1}{\sigma_j^2}(d^t - y_j^t)\frac{\partial y_j^t}{\partial \theta_j}. \tag{35}$$

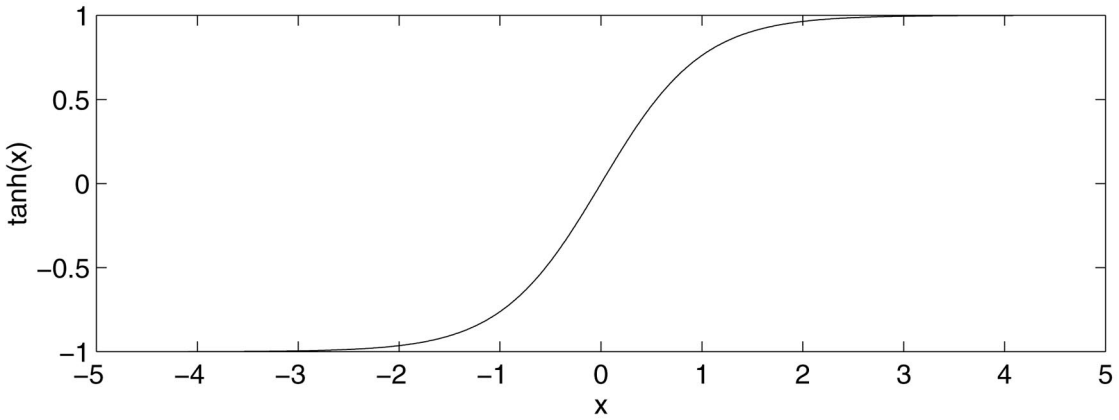

Fig. 4. The hyperbolic tangent function. The SSGE model breaks the tanh map into three segments, as shown in Fig. 5. The segments correspond to the two flat regions and the sloped region.

Notice that this equation retains the traditional weight update form for a single neural network trained on sum-squared error with a linear output unit (the appropriate linking function for the sum-squared error criterion). This update is weighted by the ratio of the β parametrized association probability and the confidence in the expert σ_j^2. In the limit that the association probabilities are binary, the gradient is nonzero only if the jth expert is appropriate for the given set of dynamics. Other expert networks remain unchanged.

The gate network has an interesting update rule, given by the following formula:

$$\frac{\partial F}{\partial \theta_g} = -\sum_{t=1}^{T} H_j^t(\beta)(1 - g_j^t)x^t. \tag{36}$$

Comparing this update rule to the one given in the standard EM algorithm (see [22]), we find that there is a difference between the update equations. Instead of obtaining a difference between the target H and the gate output g, as in the traditional EM setting, we obtain a slightly different comparison of the two values. Operationally, though, these methods produce similar results.

4.3 Derivation of the Optimal Parameters for the Linear Case

Although the SSGE model is a *nonlinear* model, it is instructive to compute the parameter updates for the linear case. This gives us an indication for how the SSGE model might behave in the nonlinear case. These equations are computed by taking the appropriate derivative of the F function, setting it equal to zero, and solving for the parameters. We give the final results of these calculations below. The expert parameters are computed as follows:

$$\frac{\partial F}{\partial \theta_j} = 0 \Rightarrow \tag{37}$$

$$\theta_j = \left[\sum_{t=1}^{T} H_j^t(\beta)x^t(x^t)^T\right]^{-1}\left[\sum_{t=1}^{T} H_j^t(\beta)d^t x^t\right]. \tag{38}$$

This equation shows that the regression parameters are the solution to a least squares problem where the input and target values are weighted by the association probability. Therefore, if the association probabilities are binary (which occurs with a large value of β), the regression parameters are solely a function of the subspace that is appropriate for the expert. There is, therefore, no sharing of the subspaces between experts.

We next derive the variance σ_j^2 for the jth expert. This is computed according to:

$$\frac{\partial F}{\partial \sigma_j^2} = 0 \Rightarrow \tag{39}$$

$$\sigma_j^2 = \frac{\sum_{t=1}^{T} H_j^t(\beta)(d^t - y_j^t)^2}{\sum_{t=1}^{T} H_j^t(\beta)}. \tag{40}$$

This equation is identical to the equation obtained for the traditional gated experts model and shows that the variance is simply the weighted sum of squares of the errors between the desired value and the predicted value for the kth expert. This result is independent of whether or not the model is linear.

The parameters for the gate, θ_{gj} are next derived, which are the parameters of the gate network for the jth output. Again, taking the appropriate partial derivative we have:

$$\frac{\partial F}{\partial \theta_{gj}} = -\sum_{t=1}^{T} H_j^t(\beta)(1 - g_j^t)x^t \tag{41}$$

$$\Rightarrow \theta_{gj} = \left[\sum_{t=1}^{T} H_j^t(\beta)x^t(x^t)^T\right]^{-1}\left[\sum_{t=1}^{T} H_j^t(\beta)x^t\right]. \tag{42}$$

In gradient descent optimization, the value of g_j^t is drawn to the value of $H_j^t(\beta)$. To see this, consider the situation where the value of $H_j^t(\beta)$ is equal to unity. In this case, the gradient is equal to zero only if g_j^t moves toward unity. Since the sum of the g's equals unity, the other values of $g_k^t, k \neq j$ are driven to zero.

4.4 Training

The SSGE is trained using the following multistep process:

- Choose initial values for the parameters of the experts, θ_1, θ_2, $\cdots$, θ_K, and the gate network, θ_g.

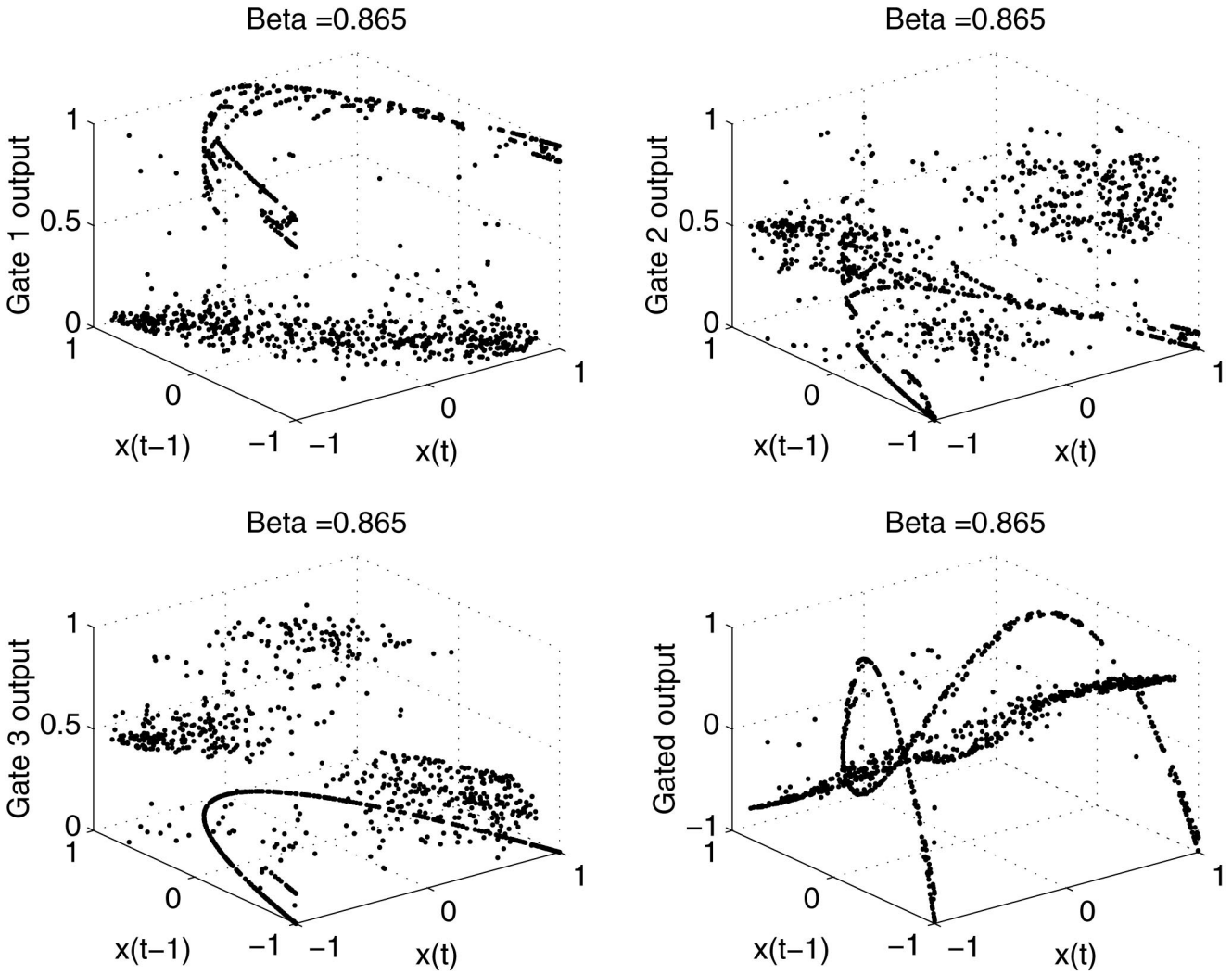

Fig. 5. Gate output for $\beta = 0.865$. For this level of segmentation granularity, the experts begin to specialize. Expert 1 models the quadratic map, and the other two experts model the tanh map. Expert 2 models the extremes in the data and expert 3 models the midrange data. The task of modelling the low range data shared by expert 2 and 3.

We set all initial variances σ_1, σ_2, $\cdots$, σ_K, to the variance of the data.
- Choose an initial value of β.
- Minimize the cost function given in (34) until overfitting on a validation set occurs.
- Increase the value of β and retrain.

5 Overview of the Experiments

We demonstrate the SSGE on a time series segmentation problem. The time series problem addresses the issues of segmenting multistationary time series discussed in Section 2, and is a computer generated series obtained by randomly switching between two nonlinear processes.

This section contains the results for the SSGE model for time series segmentation. In particular, we show the behavior of the model on a synthetic time series which exhibits a random switching between two different regimes. The SSGE model correctly identifies the subprocesses for the synthetic time series. The simulation shows:

- The nature of the segmentations that the SSGE model delivers as a function of the granularity parameter β. For low values of β, each data point is associated with each expert, indicating no specialization. For large values, each data point is associated with only one expert, indicating that the expert is overspecializing. Intermediate values produce segmentations which fall between these two extremes.
- The learning dynamics of the SSGE model. These curves include the the expected normalized mean square error (E_{NMS}) and the variances of the experts. The E_{NMS} can increase during training, thus indicating that the model is adjusting its parameters to change the segmentation. The variances characterize the predictability of the subprocesses.
- The distributions of the gate outputs as a function of β. We find that over a large range of βs, the output distributions indicate that a fixed number of experts are needed to model the time series. This number depends on the data set.

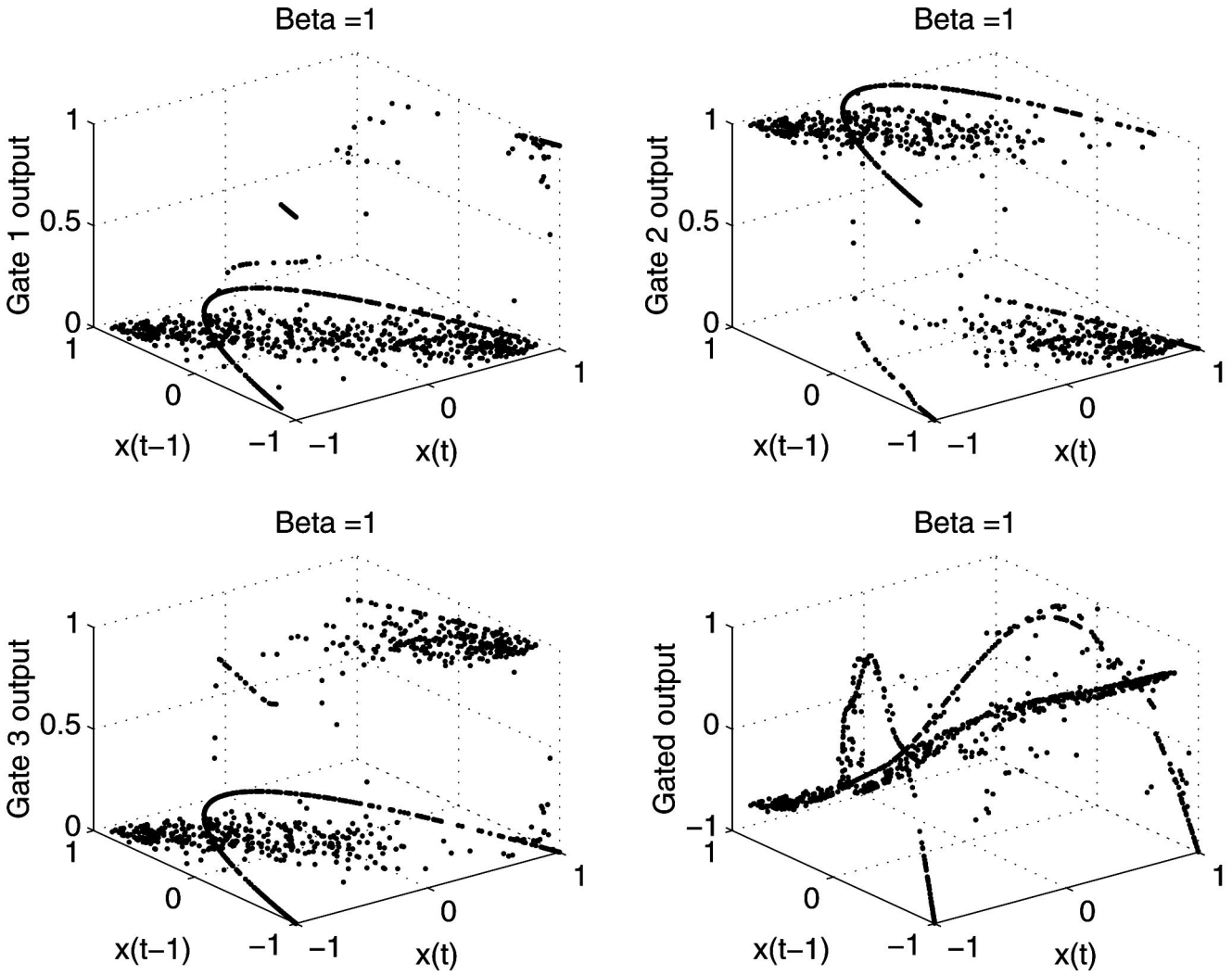

Fig. 6. Gate output for $\beta = 1$. The gate network separates the dynamics to some extent. The gate output for expert 1 shows that this expert is not used; only two experts are used. The segmentation is different from the data generating process.

The computer generated time series obeys a Markov switching process. This process is *separable*, meaning that it is possible to distinguish between the two subprocesses given the input.

6 Computer-Generated Data

6.1 Data: Mixture of Two Processes

We generated a time series which obeys the following switching process:

$$d^{t+1} = 2\,(1 - d^t)^2 - 1 \;\text{ if switch} = 1 \tag{43}$$

$$\begin{aligned} d^{t+1} &= \tanh(-1.2d^t + \varepsilon^{t+1}); \\ &\varepsilon \sim \mathcal{N}(\text{mean} = 0, \text{var} = 0.1) \;\text{ if switch} = 0. \end{aligned} \tag{44}$$

The first process is the logistic map, which exhibits deterministic chaos (low noise regime), whereas the second process is a nonlinear autoregressive (AR) process of order 1. The variance of the added noise is 0.1 which produces a relatively high noise regime. The switching dynamics is governed by a first order Markov process with transition probability 0.02. This means that on average, the process will undergo a state transition after every $\frac{1}{0.02} = 50$ time steps.

Fig. 2 indicates the nature of the subprocesses. An important difference between the two subprocesses is the noise level. The logistic map is noise-free, whereas the tanh map has injected noise. This characteristic identifies the two regimes. The second panel in the figure indicates that the subprocesses are separable given the two inputs (d^t, d^{t-1}).

6.2 Architecture and Learning

We used an SSGE model with four lagged inputs to the gate, two lagged inputs to the experts,[1] and 10 tanh hidden units. We explored other architectures which used different numbers of hidden units and found that the number of hidden units was adequate for the problem. This SSGE model had a total of three experts at its disposal. We know a priori that the "correct" solution is where the SSGE model finds the two regimes with two experts, and eliminates the remaining unnecessary expert.

1. A lagged input of order m is defined as a vector $\mathbf{x}^t = d^t, \ldots, d^{t-m}$ [21].

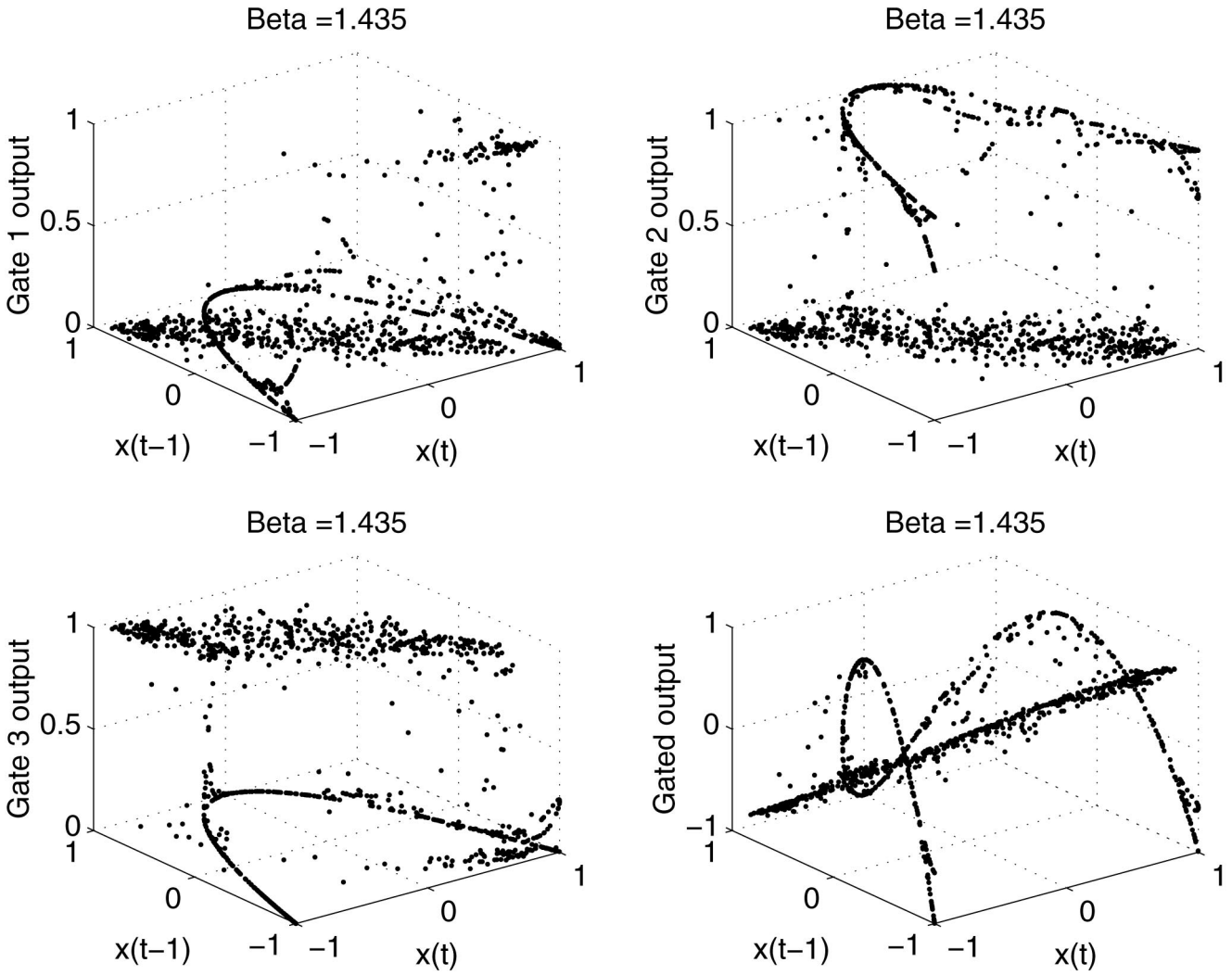

Fig. 7. Gate output for $\beta = 1.435$. For this value of β, the segmentation is virtually perfect: one expert is rarely used, one expert models the parabola, and the third expert models the tanh data.

We find that the SSGE model explores a range of possible segmentations, at the highest level, varying from three to one expert, and for a given number of experts, the SSGE model divides the time series into nontrivial but sensible components. In contrast to the results for a the same problem using the standard Gated Experts model, we obtain a variety of segmentations, whereas [22] obtain a single segmentation.

6.3 Segmentations and Analysis

We show the results for the SSGE model in the following plots. Fig. 3, Fig. 4, Fig. 5, Fig. 6, and Fig. 7 are the output of the gate network as a function of (d^t, d^{t-1}) along with the output of the entire SSGE model (gate and experts). We expect the output of the entire model to closely mimic the second panel in Fig. 2. We express the quality of the overall model in terms of the *normalized mean squared error* which is computed according to the following formula:

$$E_{\mathrm{NMS}} = \frac{\sum_{k\in\mathcal{T}}(\mathrm{observation}_k - \mathrm{prediction})^2}{\sum_{k\in\mathcal{T}}(\mathrm{observation}_k - \mathrm{mean}_{\mathcal{T}})^2}. \quad (45)$$

E_{NMS} compares the performance of the model on set $\mathcal{T}$ to simply predicting the mean on that set. For the SSGE model, we obtained a values of E_{NMS} which varied between 0.14 and 0.2. The reason for this variation is because if the segmentation is inappropriate, it may be difficult to model given the network resources. The theoretical lower bound for E_{NMS} for this example is computed below:

$$E_{\mathrm{NMS}} = \frac{\sum_{k\in\mathcal{T}}(\mathrm{observation}_k - \mathrm{prediction})^2}{\sum_{k\in\mathcal{T}}(\mathrm{observation}_k - \mathrm{mean}_{\mathcal{T}})^2} \quad (46)$$

$$= \frac{(0.5)\times(0) + (0.5)\times(0.1)}{0.45} \quad (47)$$

$$= 0.11. \quad (48)$$

We obtain the expression in (47) because the transition probabilities are symmetric so the system will, on average, spend half its time in the quadratic map, which has no noise, so the SSGE model should have a perfect approximation with zero error, and the other half of its time in the tanh map. The tanh map, unlike the quadratic map has added noise with variance of 0.1, so the best performance any approximator can produce will have a

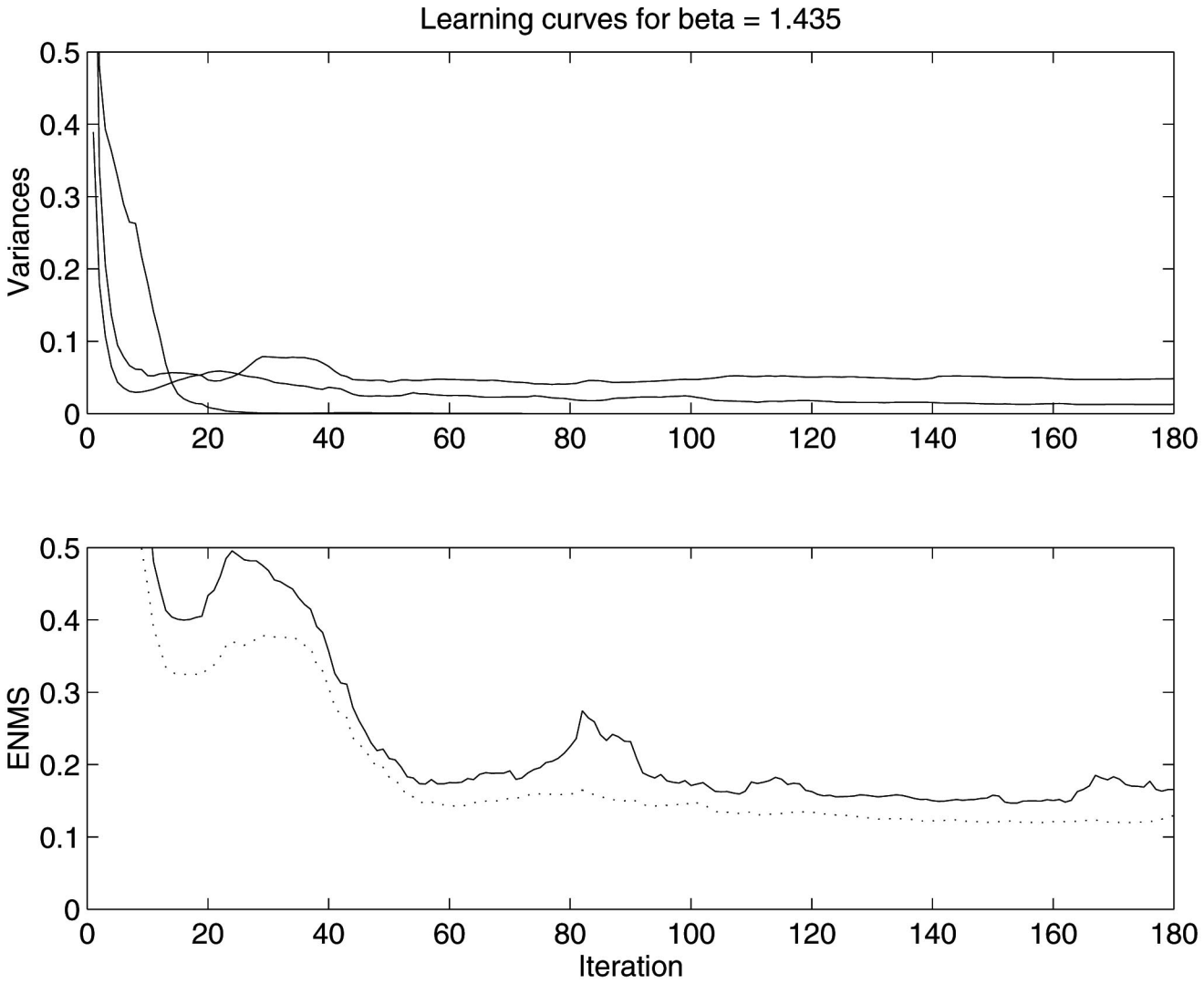

Fig. 8. Learning curves for $\beta = 1.435$ on the quadratic-tanh data. The curves on the top panel indicate the variances of the experts, while the lower panel contains the E_{NMS} for the training set (dotted line) and the test set (solid line). The E_{NMS} increases around iteration 20 and again at iteration 90.

variance of 0.1. The denominator of the expression is the variance of a large (10,000 data points) sample with half the samples from the quadratic map and the other half from the tanh map.

Fig. 3 shows the output of the gate network with $\beta = 0.01$. All plots shown here are computed on the test set (out-of-sample predictions). The SSGE model produces a segmentation where each data point is equally associated with each expert. This segmentation treats the entire data set as a single series.

An interesting situation occurs as we increase the value of β. Fig. 5 shows the gate output for $\beta = 0.865$. The SSGE model is producing a segmentation which uses all three experts. The quadratic map is separated from the tanh map, but the tanh map is divided into three regions: the extreme where the inputs are near $(1, 1)$, the midrange, and the low range where the inputs are near the origin. One expert models the extreme, one models the midrange, and the two experts combined model the low range. With this segmentation, the noisy regime is divided into the three regions which correspond to the three segments of the hyperbolic tangent curve (see Fig. 4.)

For $\beta = 1$, two experts are used which corresponds to the standard gated experts model. Thus, this figure serves as an illustration of the gated experts model's performance on this data set. Fig. 6 shows the gate outputs for this case. This segmentation is typical for the standard gated experts model and indicates that the separation of the dynamics is not appopriate. For larger values of β, we obtain different segmentations.

We obtain a perfect segmentation of the series for a value of $\beta \approx 1.4$ as shown in Fig. 7. This segmentation is characterized by using only two experts, devoting one expert solely to the quadratic map, and devoting the other to the tanh map. We explored the neighborhood of $\beta = 1.4$ and found that this segmentation occurs in a small neighborhood of this value (from about 1.35 to 1.45). This segmentation, although perfect, does not seem to be robust in variations in β. The SSGE model gives the user the ability to choose a particular segmentation from a variety of segmentations.

6.4 Learning Curves

The learning curves are included here because they reveal important features about the learning dynamics of the SSGE model. The variances show the degree of specialization of the experts and the search for the regime to specialize in. The normalized mean squared error (E_{NMS}) shows the predictive performance of the SSGE model. We find that the SSGEv model can trade predictive performance for segmentation performance.

We now show the variances and the E_{NMS} for $\beta = 1.435$. Fig. 8 shows that the model drives one expert to a low variance (on the order of 10^{-6}). The lower panels of this figure contain the E_{NMS} as a function of the training time.

Fig. 8 shows the interesting situation where the E_{NMS} briefly increases during training. As Weigend et al. [22] points out, these increases are due to the trade-off between the segmentation and the predictive power. The final E_{NMS} converges to 0.14, just above the theoretical lower bound.

7 Conclusions

We have shown that Scale-Sensitive Gated Experts perform feature refinement for complex nonlinear regression problems. The feature refinement is governed by a scale parameter which naturally arises in the model derivation. Each local regression model models different features of the regression surface while the gate network partitions the input-output space to a level of granularity that is set by the scale parameter.

Acknowledgments

The authors thank Noureddine Kermiche, Shanming Shi, Jens Timmer, Inder Batra, and Steve Waterhouse for valuable comments and suggestions.

References

[1] M. Basseville and I.V. Nikiforov, *Detection of Abrupt Changes: Theory and Application.* Prentice Hall, 1993.
[2] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* New York: John Wiley, 1991.
[3] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* Wiley, 1973.
[4] R. Durbin and D. Willshaw, "An Analogue Approach to the Travelling Salesman Problem Using an Elastic Net Method," *Nature,* pp. 689-691, 1987.
[5] C. Fancourt and J. Principe, "A Neighborhood Map of Competing One Step Predictors for Piecewise Segmentation and Identification of Time Series," *Proc Int'l Conf. Neural Netorks,* 1996.
[6] N. Gershenfeld, "Nonlinear Inference and Cluster-Weighted Modeling," *Proc. 1995 Florida Workshop Nonlinear Astronomy,* vol. 1, pp. 1-6, 1995.
[7] S. Guiasu, *Information Theory with Applications.* McGraw-Hill, 1977.
[8] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation.* Reading, Mass.: Addison-Wesley, 1991.
[9] R.A. Jacobs and M.I. Jordan, "Learning Piecewise Control Strategies in a Modular Network Architecture," *IEEE Trans. Systems, Man, and Cybernetics,* 1993.
[10] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation,* vol. 3, pp. 79-87, 1991.
[11] M.I. Jordan and R.A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation,* vol. 6, pp. 181-214, 1994.
[12] P. McCullagh and J.A. Nelder, *Generalized Linear Models.* London. Chapman and Hall, 1989.
[13] K. Pawelzik, J. Kohlmorgen, and K.-R. Müller, "Annealed Competition of Experts for a Segmentation and Classification of Switching Dynamics," *Neural Computation,* vol. 8, no. 2 pp. 340-356, 1996.
[14] R.E. Quandt, "The Estimation of the Parameters of a Linear Regression System Obeying Two Separate Regimes," *J. Am. Statistical Assoc.,* pp. 873-880, 1958.
[15] C.R. Rao, *Linear Statistical Inference and its Applications.* New York: John Wiley and Sons, 1965.
[16] K. Rose, E. Gurewitz, and G.C. Fox, "Statistical Mechanics and Phase Transitions in Clustering, *Physical Rev. Letters,"* vol. 65, no. 8, pp. 945-948, 1990.
[17] D.E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, "Backpropagation: The Basic Theory," *Backpropagation: Theory, Architectures, and Applications,* Y. Chauvin and D.E. Rumelhart, eds., pp. 1-34, Hillsdale, N.J.: Lawrence Erlbaum Assoc., 1995.
[18] S. Shi, "Modeling the Temporal Structure of Time with Hidden Markov Experts," PhD thesis, Dept. of Computer Science, Univ. of Colorado, 1998.
[19] L.W. Swokowski, *Calculus with Analytic Geometry.* Prindle Weber and Schmidt, 1984.
[20] F. Takens, "Detecting Strange Attractors in Turbulence," *Dynamical Systems and Turbulence,* D.A. Rand, and L.S. Young, eds. *Lecture Notes in Mathematics,* vol. 898, pp. 366-381, Springer, 1981.
[21] *Time Series Prediction: Forecasting the Future and Understanding the Past.* A.S. Weigend and N.A. Gershenfeld, eds., Reading, Mass.: Addison-Wesley, 1994.
[22] A.S. Weigend, M. Mangeas, and A.N. Srivastava, "Nonlinear Gated Experts for Time Series: Discovering Regimes and Avoiding Overfitting," *Int'l J. Neural Systems,* vol. 6, pp. 373-399, 1995.
[23] Y. Wong, "Clustering Data by Melting," *Neural Computation,* vol. 5, pp. 89-104, 1993.

Ashok N. Srivastava received his PhD degree in electrical engineering from the University of Colorado, Boulder, in 1996. He is chief technologist of the Deep Computing Consulting Group at IBM, where he creates data mining algorithms for time series forecasting in the finance, telecommunications, and manufacturing industries. Before joining IBM, he was a research scientist at the NASA Ames Research Center, where he developed methods in fault forecasting and detection and time series segmentation. He has over 30 publications to his credit, including editorship of one book, and authorship of two chapters in a textbook.

Renjeng Su received the BSChE degree from Chenkung University, Taiwan, in 1972, and the DSc degree in system science and mathematics from Washington University, St. Louis, in 1980. He is presently a faculty member in electrical and computer engineering at the University of Colorado, Boulder and is also with the Colorado Center for Information Storage. He is a fellow of the IEEE.

Andreas S. Weigend received his PhD degree from Stanford University in 1991, worked on text mining at Xerox PARC (Palo Alto Research Center), coorganized the Time Series Competition at the Santa Fe Institute, was an assistant professor of computer science and cognitive science at the University of Colorado, Boulder, and is currently associate professor of information systems at New York University's (NYU) Stern School of Business. He has published more than 100 articles in scientific journals, books, and conference proceedings, and has coedited five books.